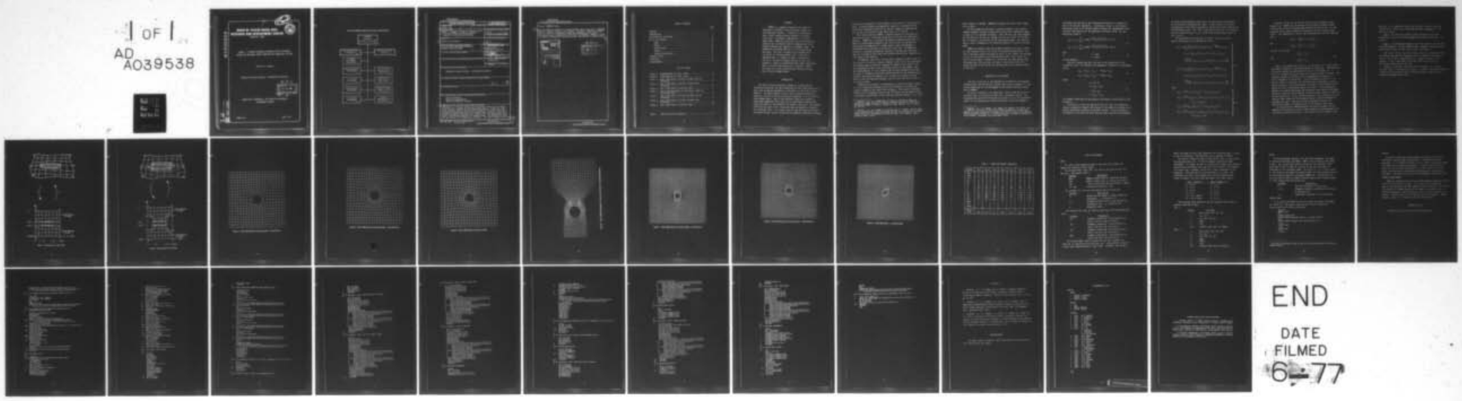
AD-A039 538 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 12/1
NUMESH: A COMPUTER PROGRAM TO GENERATE FINITE-DIFFERENCE MESHES--ETC(U)
MAR 77 R M COLEMAN

UNCLASSIFIED CMLD-77-05 NL

1 OF 1
AD
A039538



END
DATE
FILMED
6-77

ADA039538

# DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

## NUMESH: A COMPUTER PROGRAM TO GENERATE FINITE-DIFFERENCE MESHES FOR ARBITRARY DOUBLY-CONNECTED TWO-DIMENSIONAL REGIONS

Roderick M. Coleman

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.



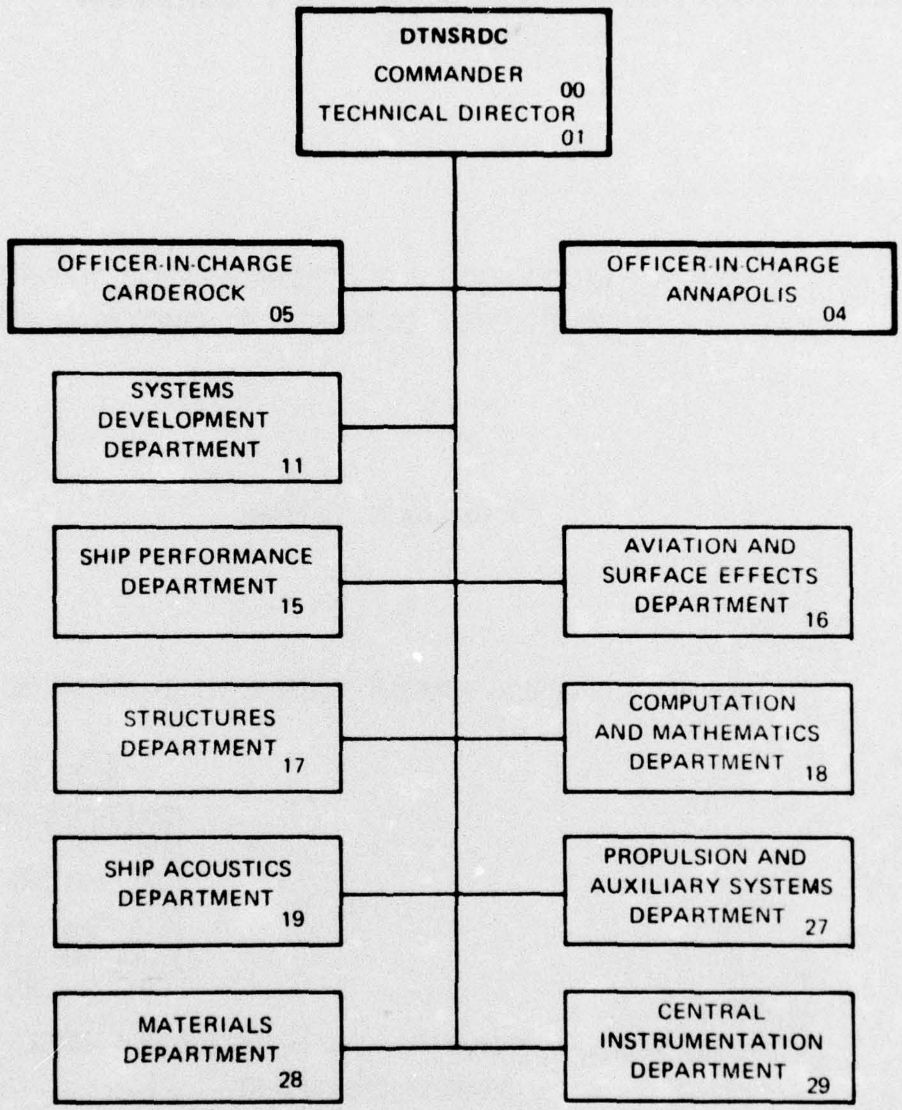
COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT
DEPARTMENTAL REPORT



MARCH 1977

CMLD-77-05

# MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER: 14 CMLD-77-05 | 2. GOVT ACCESSION NO. |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE (and Subtitle): 6 NUMESH: A Computer Program to Generate Finite-Difference Meshes for Arbitrary Doubly-Connected Two-Dimensional Regions. | 5. TYPE OF REPORT & PERIOD COVERED: 9 Final rept. |
| | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s): 10 Roderick M. Coleman | 8. CONTRACT OR GRANT NUMBER(s): 16 SR01402 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship R&D Center, Bethesda, Maryland 20084 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 17 SR 014 02 01, Work Unit 1-1843-015 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS | 12. REPORT DATE: 11 March 1977 |
| | 13. NUMBER OF PAGES: 35 12 36p. |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | 15. SECURITY CLASS. (of this report): UNCLASSIFIED |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

406 847

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite Difference
Numerical Transformation
Partial Differential Equation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

NUMESH is a computer program for the numerical generation of boundary-fitted coordinate systems for two-dimensional regions. A numerical transformation maps a doubly-connected region bounded by arbitrary curves onto a rectangular computational field with a square mesh. Numerical solutions of partial differential equations may be obtained from the computational field without interpolation regardless of boundary shape. The mathematical procedure and the use of the program are described. Two types of meshes can be next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20. ABSTRACT (cont.)

cont

produced which are useful for solving problems such as flow past a submerged body in a channel or under a free surface. The use of interactive graphics permits a mesh to be generated, viewed, and regenerated with slight alterations, until a suitable mesh definition is obtained. Meshes are shown for several configurations involving circular and arbitrarily-shaped bodies.

## TABLE OF CONTENTS


| | Page |
|---|---|
| ABSTRACT | 1 |
| INTRODUCTION | 1 |
| DESCRIPTION OF THE METHOD | 3 |
| USE OF THE PROGRAM | 18 |
| Input | 18 |
| Output | 20 |
| Control Cards | 20 |
| Graphics | 21 |
| Computer Requirements | 21 |
| PROGRAM LISTING | 21 |
| REFERENCES | 31 |
| ACKNOWLEDGMENT | 31 |


## LIST OF FIGURES


| | | |
|---|---|---|
| Figure 1. | Transformation for Type 1 Mesh | 8 |
| Figure 2. | Transformation for Type 2 Mesh | 9 |
| Figure 3. | Type 1 Mesh About a Circular Cylinder with No Attraction | 10 |
| Figure 4. | Type 1 Mesh About a Circular Cylinder With Attraction | 11 |
| Figure 5. | Type 1 Mesh About an Arbitrary Body | 12 |
| Figure 6. | Type 1 Mesh Developed for Flow About a Body in a Constricted Tube | 13 |
| Figure 7. | Type 2 Mesh About a Circular Cylinder with No Attraction | 14 |
| Figure 8. | Type 2 Mesh About a Circular Cylinder with Attraction | 15 |
| Figure 9. | Type 2 Mesh About an Arbitrary Body | 16 |

| | | |
|---|---|---|
| Table 1 - | Input and Control Parameters | 17 |

## ABSTRACT

NUMESH is a computer program for the numerical generation of boundary-fitted coordinate systems for two-dimensional regions. A numerical transformation maps a doubly-connected region bounded by arbitrary curves onto a rectangular computational field with a square mesh. Numerical solutions of partial differential equations may be obtained from the computational field without interpolation regardless of boundary shape. The mathematical procedure and the use of the program are described. Two types of meshes can be produced which are useful for solving problems such as flow past a submerged body in a channel or under a free surface. The use of interactive graphics permits a mesh to be generated, viewed, and regenerated with slight alterations, until a suitable mesh definition is obtained. Meshes are shown for several configurations involving circular and arbitrarily-shaped bodies.

## INTRODUCTION

The value of finite-difference schemes for solving partial differential equations has been demonstrated by their wide-spread use in many areas of applied mathematics, particularly in the field of fluid dynamics. The usual problem is one in which a partial differential equation or system of partial differential equations is to be solved on some bounded region. Many of the available schemes are limited by complicated geometries which give rise to difficulties stemming from inaccurate numerical representation of boundary conditions. The boundary conditions are usually best represented when the boundaries themselves are coordinate lines. As a result, much use has been made of "natural" coordinate systems such as cylindrical and spherical coordinates, although

the practical usefulness of such systems is restricted to a few specialized cases. A general method was needed which would give a *boundary-fitted* curvilinear coordinate system for any specific geometric configuration, such as that of an arbitrary body shape under a free surface.

This report describes NUMESH, a computer program for the numerical generation of such a coordinate system for an arbitrary two-dimensional geometry. NUMESH uses a procedure developed by Thompson[1] which leads to a numerical transformation mapping a set of grid points in the physical region under consideration to a square mesh so that each boundary is coincident with a grid line or portion of a grid line. The numerical solution of a partial differential equation in the physical region may then be calculated on the square mesh without interpolation, regardless of body shape or mesh spacing.

In Thompson's early work,[1] he describes a mapping in which a body in the interior of a physical region is mapped to one side of the rectangular transformed plane and the outer boundary is mapped to the opposite side. The remaining two sides of the calculating region are the images, under the transformation, of a cut running from the outer boundary to the body. The type of coordinate system thus created has many of the characteristics of polar coordinates and appears to be suitable for problems in which there is only one important boundary, such as fluid flow about a body in an infinite fluid.

Thompson also presents[2] a transformation in which the body is mapped to a slit within the rectangle and the outer boundary of the physical region is mapped to the sides of the rectangle. This type of coordinate system is more rectangular than the one first described and seems suited to problems in which there is more than one important boundary, e.g., flow

---

1 Thompson, J.F., F.C. Thames and C.W. Mastin, "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Bodies," Journal of Comp. Physics, vol. 15 (1974), pp. 299-319.

2 Thompson, J.F., F.C. Thames, C.W. Mastin and S.P. Shanks, "Use of Numerically Generated Body-Fitted Coordinate Systems for Solution of the Navier-Stokes Equations," Proceedings of AIAA 2nd Comp. Fluid Dynamics Conf., Hartford, Conn. (1975).

about a body in a channel. NUMESH can produce this type of mesh, known here as Type 1.

For the problem of flow about a hydrofoil beneath a free surface, Thompson suggests[3] another configuration in which the transformed region is made up of two adjacent rectangles. Here the body, the outer boundary, the free-surface, and the cut running from the free surface to the body are all mapped to the exterior of the transformed region. The resulting mesh is used by Thompson for the hydrofoil problem in a fluid of infinite depth.

NUMESH can produce a new type of mesh, referred to as Type 2, which has some characteristics of all three of the above configurations. The Type 2 coordinate system, described in detail in this report, was designed in part for the problem of an arbitrary body under a free surface in a fluid of finite depth. Because it is polar near the inner boundary and rectangular near the outer one, the Type 2 system allows the user to refine the mesh in the vicinity of the body with little change elsewhere in the field.

## DESCRIPTION OF THE METHOD

The brief discussion of the mathematical formulation of the method, presented here, and the notation, follow those given by Thompson[1,2,3] and provide background for the development of the finite difference scheme used in NUMESH.

We wish to transform a two-dimensional, doubly-connected region of arbitrary shape in the physical plane into a rectangular region. The transformation functions from the physical plane $(x,y)$ to the transformed plane $(\xi,\eta)$ are generated by solving an elliptic system with boundary

---

[3] Thompson, J.F., F.C. Thames, J.K. Hodge, S.P. Shanks, R.N. Reddy, and C.W. Mastin, "Solutions of the Navier-Stokes Equations in Various Flow Regimes on Fields Containing Any Number of Arbitrary Bodies Using Boundary-Fitted Coordinate Systems," V. International Conf. on Numerical Methods in Fluid Dynamics, Enschede, The Netherlands (1976).

conditions such that one of the transformed coordinates is constant on each of the physical boundaries. A convenient choice for the elliptic generating system is the Poisson equation because the inhomogeneous terms allow for some control over the generated coordinate system. Let the generating system be

$$\xi_{xx} + \xi_{yy} = \sum_{i=1}^{N} C_i \exp\left\{-\sqrt{(\xi-\xi_i)^2+(\eta-\eta_i)^2}\right\} \equiv P(\xi,\eta)$$

$$\eta_{xx} + \eta_{yy} + \sum_{i=1}^{N} D_i \exp\left\{-\sqrt{(\xi-\xi_i)^2+(\eta-\eta_i)^2}\right\} \equiv Q(\xi,\eta) \tag{1}$$

with

$$\xi = \text{const.}$$

or $$\eta = \text{const.} \tag{2}$$

on each boundary.

Since all computations are to be done on the square mesh in the transformed plane, the dependent and independent variables are interchanged giving

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(Px_\xi + Qx_\eta)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(Py_\xi + Qy_\eta) \tag{3}$$

where

$$\alpha = x_\eta^2 + y_\eta^2$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^2 + y_\xi^2 \tag{4}$$

$$J = x_\xi y_\eta - x_\eta y_\xi$$

The boundary conditions are the physical coordinates of mesh points on the boundaries.

The ability to alter the spacing of the grid lines in the physical plane is necessary to improve the accuracy of the numerical solution of the partial differential equation of ultimate interest. Since the physical coordinates of the points at which the grid lines intersect the boundaries

are input to the procedure, some control of spacing near the boundaries can be achieved by adjusting this input. Further control of the mesh configuration in the field is achieved by varying the P and Q functions in the generating system. The $\xi$ and $\eta$ lines may be attracted to or repelled from the specified points $(\xi_i, \eta_i)$ in the field through proper choice of the $C_i$ and $D_i$.

The approximation of Equations (3) and (4) using second-order, central differences for all derivatives yields

$$\left.\begin{aligned}
x_{i,j} &= \frac{1}{2(\alpha_{i,j}+\gamma_{i,j})}\Big\{\alpha_{i,j}(x_{i,j+1}+x_{i,j-1}) - \frac{\beta_{i,j}}{2}(x_{i-1,j+1} \\
&\quad -x_{i-1,j}-x_{i+1,j+1}+x_{i+1,j-1}) + \gamma_{i,j}(x_{i-1,j}+x_{i+1,j}) \\
&\quad + \frac{J^2_{i,j}}{2}\Big[P_{i,j}(x_{i,j+1}-x_{i,j-1})+Q_{i,j}(x_{i-1,j}-x_{i+1,j})\Big]\Big\} \\
y_{i,j} &= \frac{1}{2(\alpha_{i,j}+\gamma_{i,j})}\Big\{\alpha_{i,j}(y_{i,j+1}+y_{i,j-1}) - \frac{\beta_{i,j}}{2}(y_{i-1,j+1} \\
&\quad -y_{i-1,j}-y_{i+1,j+1}+y_{i+1,j-1}) + \gamma_{i,j}(y_{i-1,j}+y_{i+1,j}) \\
&\quad + \frac{J^2_{i,j}}{2}\Big[P_{i,j}(y_{i,j+1}-y_{i,j-1})+Q_{i,j}(y_{i-1,j}-y_{i+1,j})\Big]\Big\}
\end{aligned}\right\} \quad (5)$$

where

$$\left.\begin{aligned}
\alpha_{i,j} &= 1/4[(x_{i-1,j}-x_{i+1,j})^2 + (y_{i-1,j}-y_{i+1,j})^2] \\
\beta_{i,j} &= 1/4[(x_{i,j+1}-x_{i,j-1})(x_{i-1,j}-x_{i+1,j})+(y_{i,j+1}-y_{i,j-1}) \\
&\quad (y_{i-1,j}-y_{i+1,j})] \\
\gamma_{i,j} &= 1/4[(x_{i,j+1}-x_{i,j-1})^2 + (y_{i,j+1}-y_{i,j-1})^2] \\
J_{i,j} &= 1/4[(x_{i,j+1}-x_{i,j-1})(y_{i-1,j}-y_{i+1,j})-(x_{i-1,j}-x_{i+1,j}) \\
&\quad (y_{i,j+1}-y_{i,j-1})]
\end{aligned}\right\} \quad (6)$$

Equations (5) and (6) may then be solved on the rectangular transformed field. NUMESH uses an accelerated Gauss-Seidel iteration scheme to obtain this solution, although other methods may work as well. R, the relaxation factor used to speed the convergence, must be chosen between 0 and 2. The computation is said to have converged *after the* $k^{th}$ *iteration* when

$$\left| x^k_{i,j} - x^{k-1}_{i,j} \right| < \varepsilon_1 \left| x^k_{i,j} \right|$$

and (7)

$$\left| y^k_{i,j} - y^{k-1}_{i,j} \right| < \varepsilon_1 \left| y^k_{i,j} \right|$$

for all i,j such that

$$\left| x^k_{i,j} \right| > \varepsilon_2$$

and (8)

$$\left| y^k_{i,j} \right| > \varepsilon_2$$

There is an optimum value of R, $R_0$, in the range 0 to 2, which causes the scheme to converge in the minimum number of iterations. Experimentation with various meshes has shown that this value is often about 1.6. A choice of R greater than $R_0$ may cause the iteration scheme to diverge. The iteration procedure is halted after the computation has converged or after a maximum of 50 iterations. If the convergence criterion has not been met after 50 iterations, the calculation may be converging slowly or may be diverging. The printed output, described in the following section, should be an aid in determining which is the case. If the computation appears to be slowly converging, the relaxation factor could be increased. Additional iterations can be done by using the current results as input for another program run. If the calculation appears to be diverging, all user-supplied input, especially the relaxation factor, should be examined to determine the cause. Available interactive graphics routines are particularly helpful in locating errors in the boundary input data.

As stated earlier, NUMESH generates two types of boundary-fitted meshes when the physical region is doubly-connected, being bounded by two arbitrary non-intersecting curves. The exterior curve is mapped to the

perimeter of the computation region and the interior curve, hereafter referred to as a body, as in fluid flow problems, is mapped to a slit within the region.

Type 1: All grid lines connect either two points on the outer boundary or a point on the outer boundary and a point on the body. See Figure 1.

Type 2: A predetermined number of grid lines encircle the body and close on themselves. Special computer code is needed to calculate grid points near the body for this type of mesh. See Figure 2.

Both types of meshes are suitable when accuracy is needed near both the exterior and interior boundaries such as in problems involving the flow of a fluid past a submerged body in a channel or under a free surface. A Type 2 mesh seems to be useful when high accuracy is particularly important near the body.

Examples of the Type 1 mesh are given by Figures 3, 4, 5, and 6. The Type 2 mesh is illustrated by Figures 7, 8, and 9. The input parameters discussed in the following section, which were used to generate these examples, are summarized in Table 1.

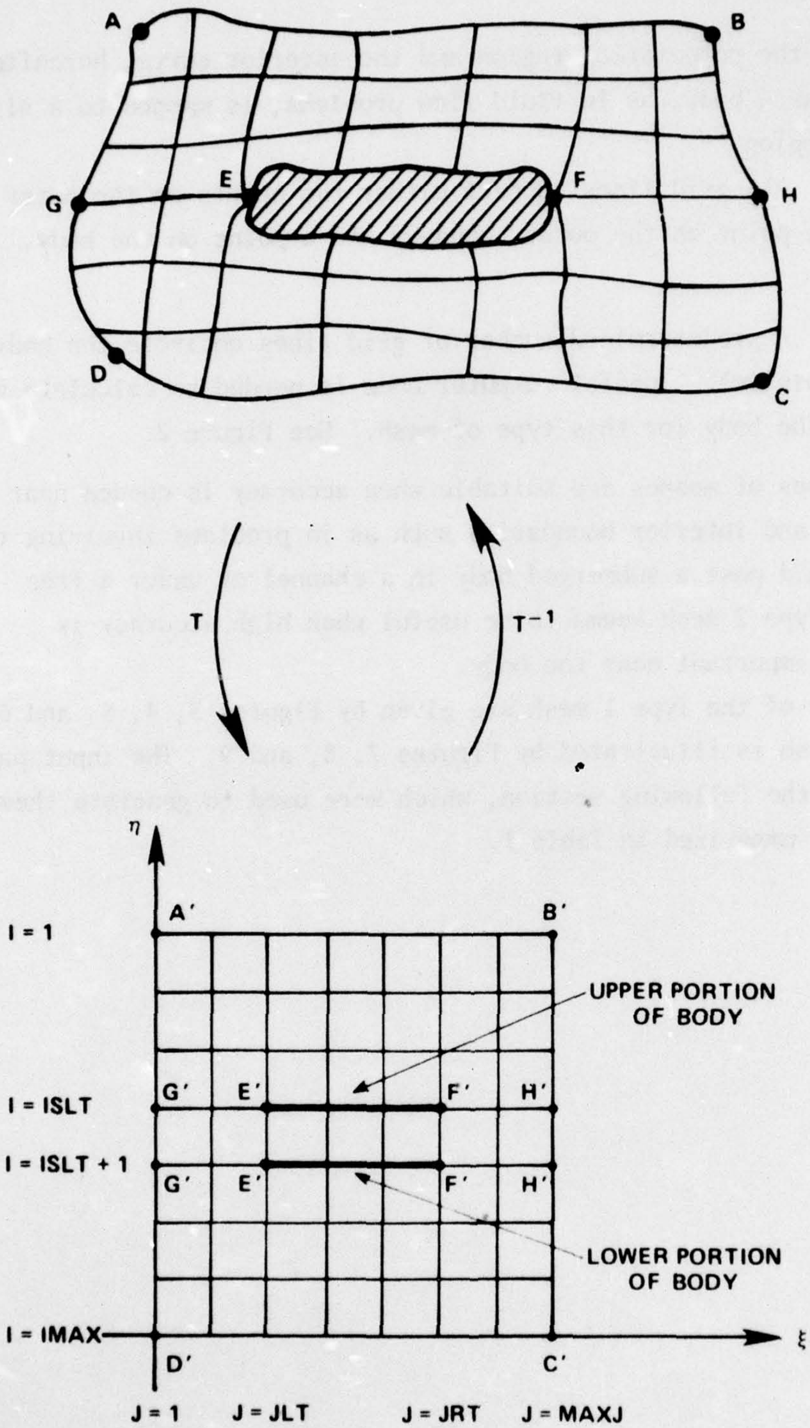



**Figure 1 Transformation For Type 1 Mesh**

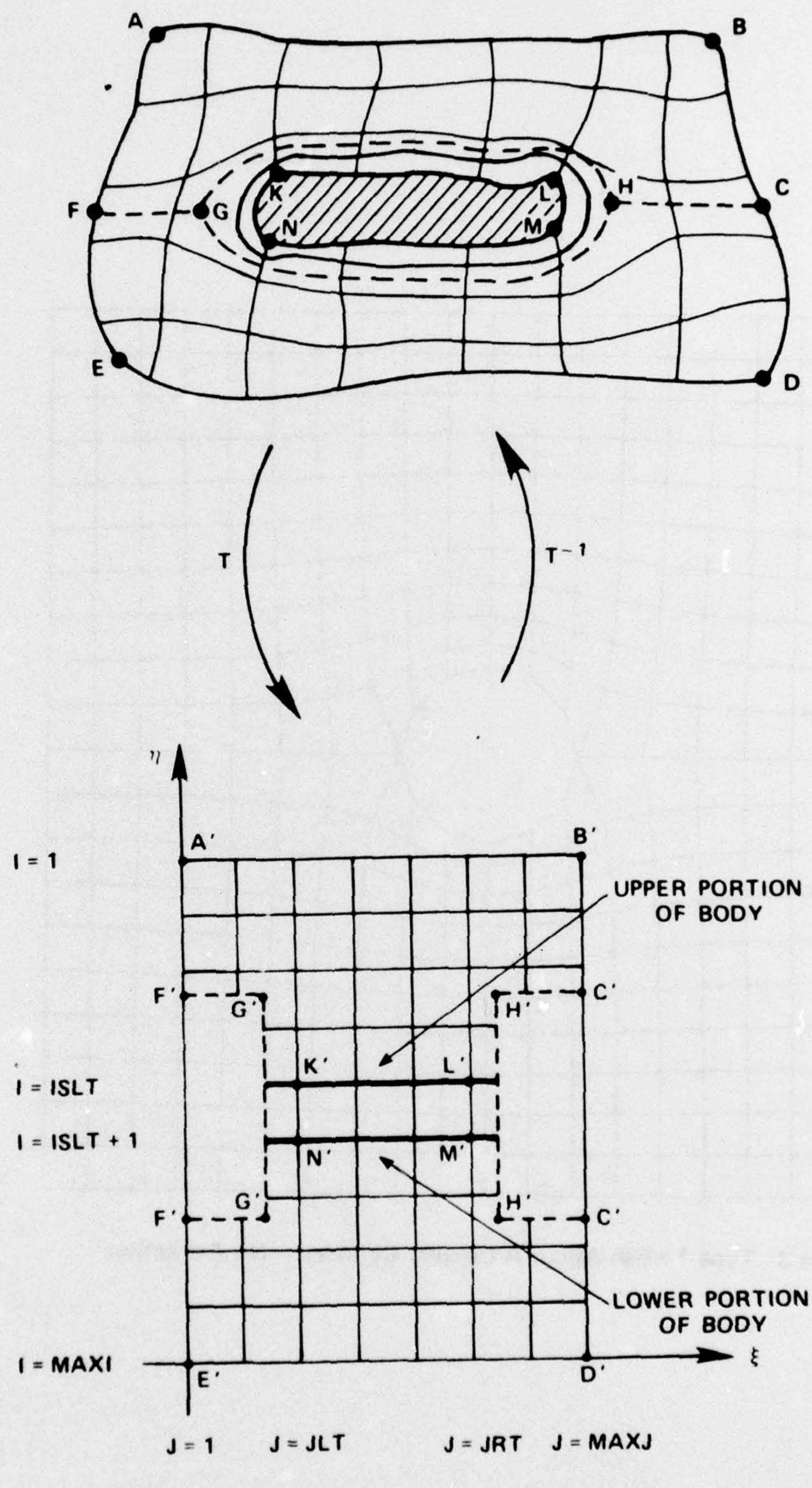



Figure 2 Transformation For Type 2 Mesh

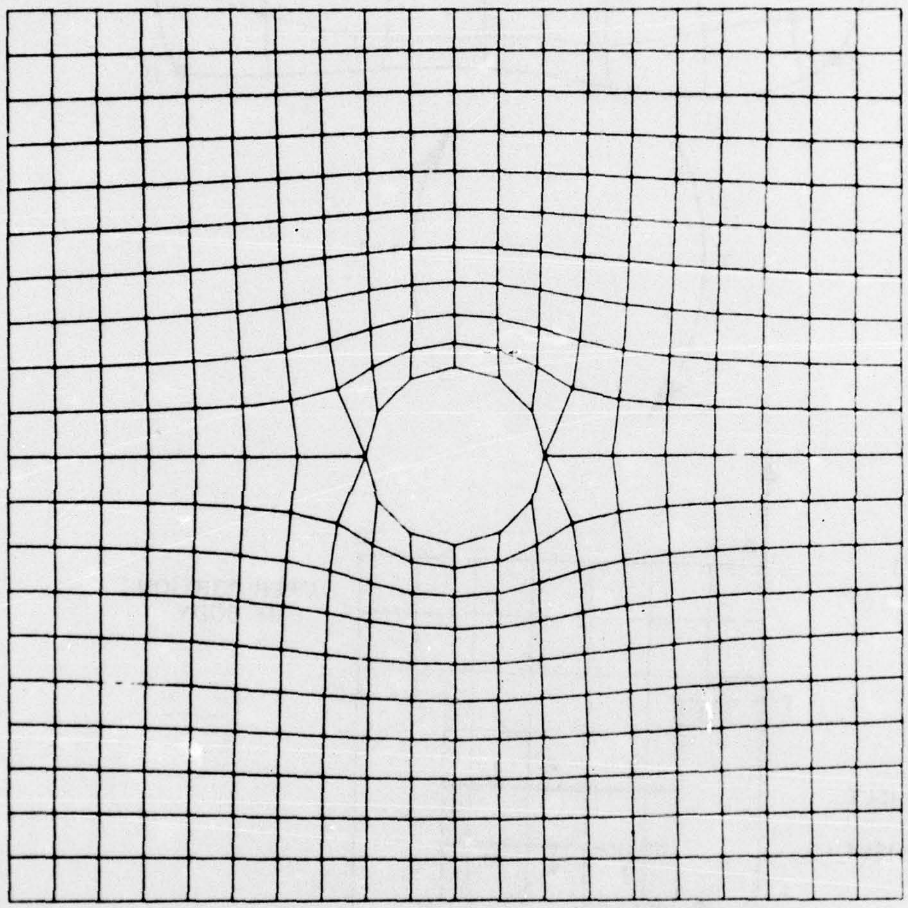


**Figure 3 Type 1 Mesh About A Circular Cylinder – No Attraction**

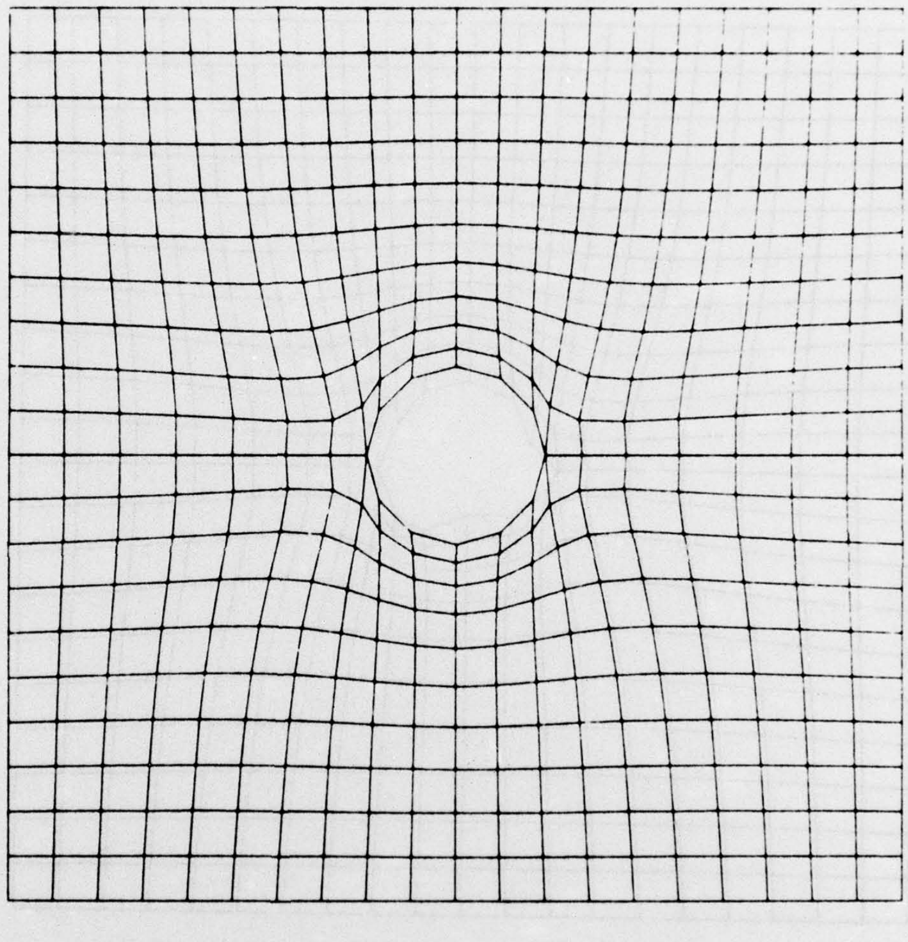


**Figure 4 Type 1 Mesh About A Circular Cylinder – With Attraction**

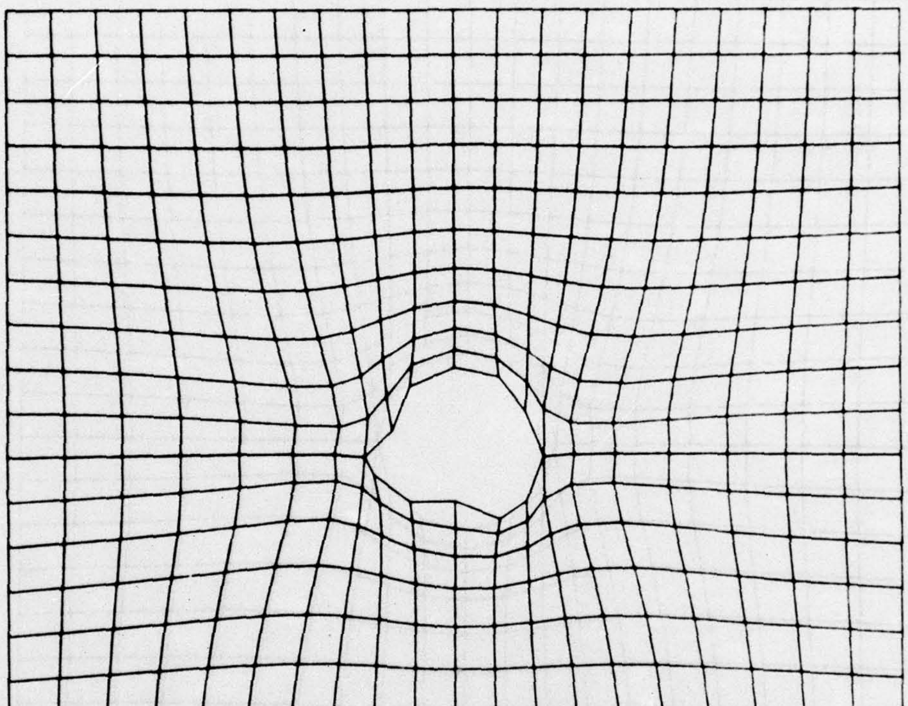


**Figure 5 Type 1 Mesh About An Arbitrary Body**

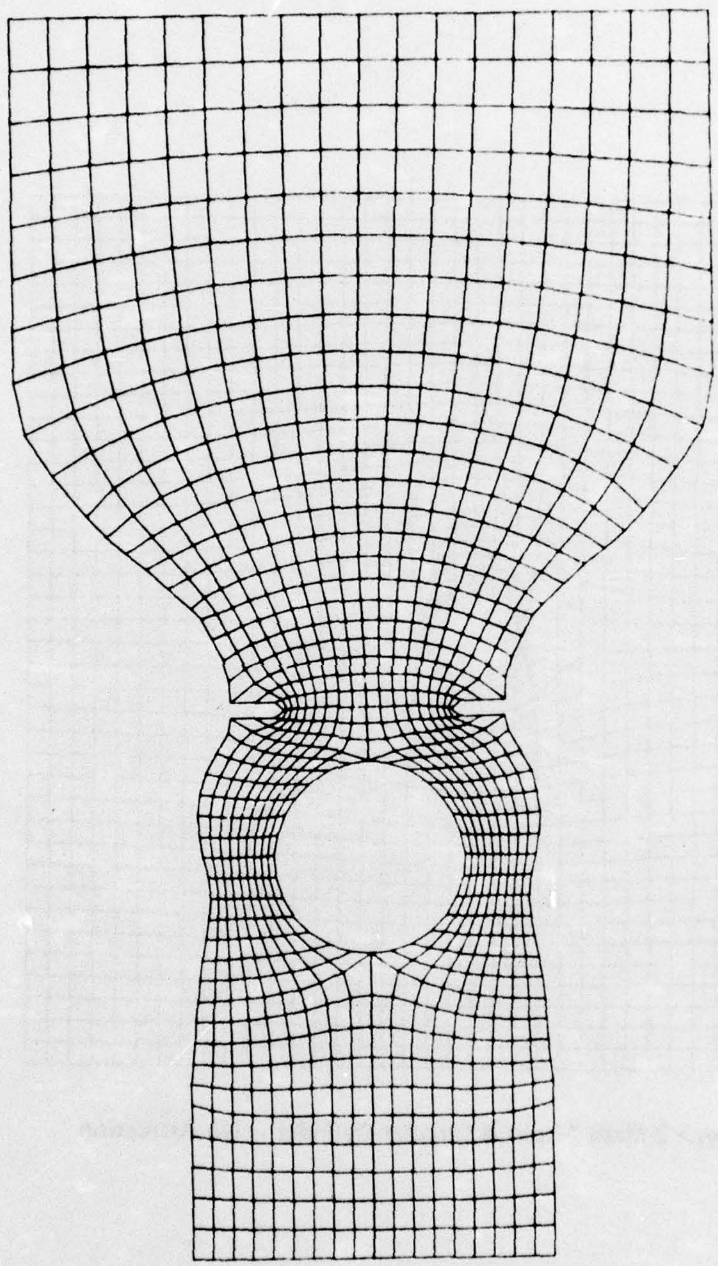


Figure 6  Type 1 Mesh Developed For Flow About A Body In A Constricted Channel

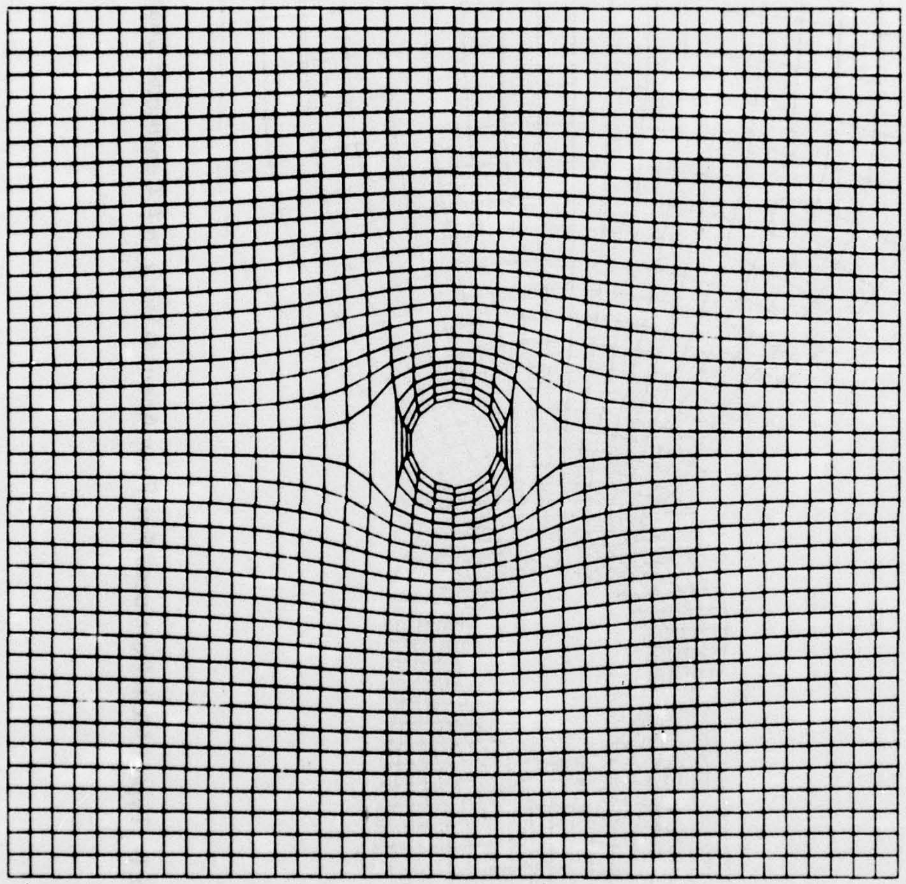


Figure 7 Type 2 Mesh About A Circular Cylinder – No Attraction

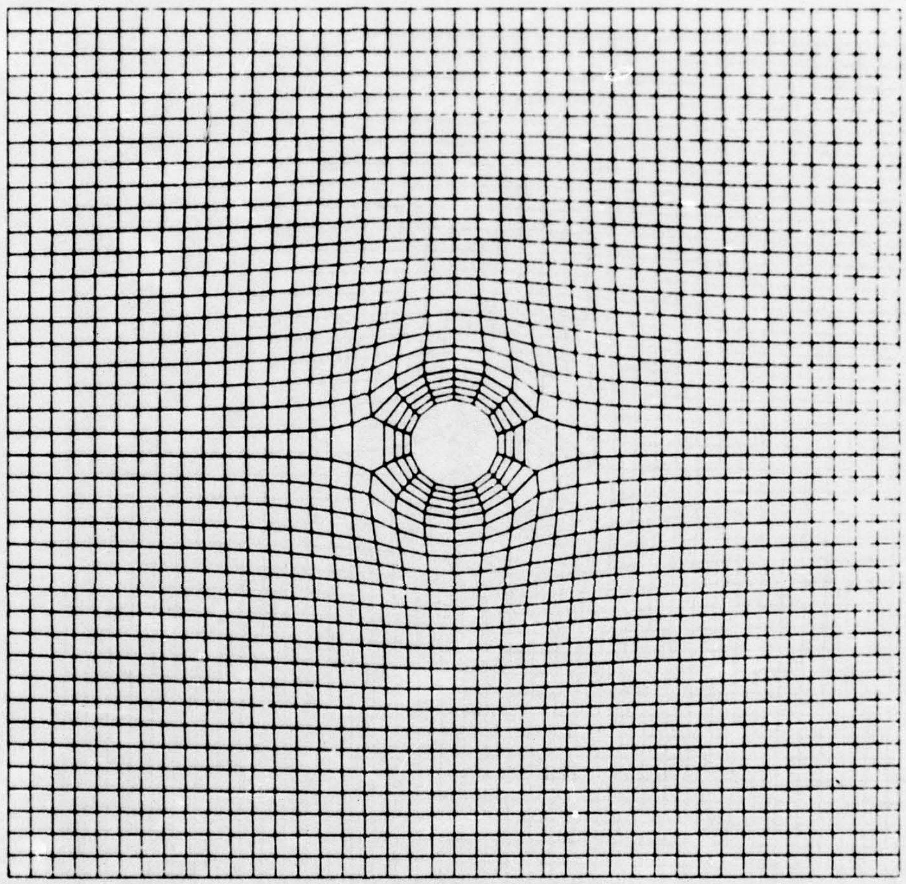


**Figure 8  Type 2 Mesh About A Circular Cylinder – With Attraction**

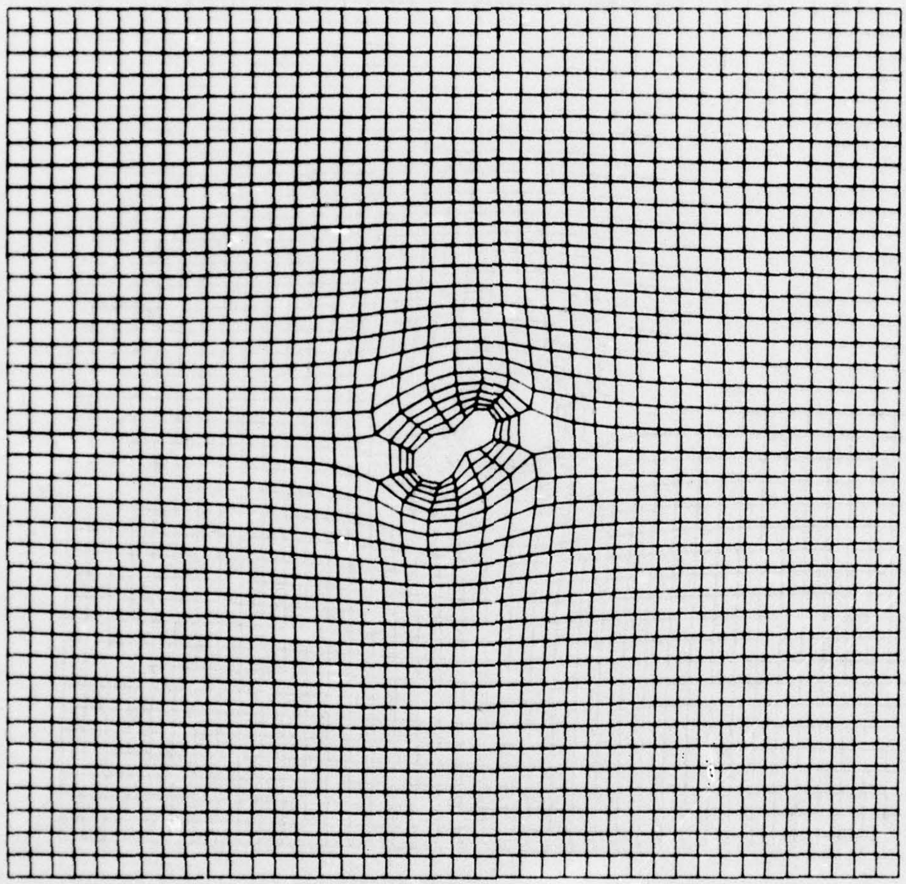


**Figure 9   Type 2 Mesh About An Arbitrary Body**

TABLE 1 - INPUT AND CONTROL PARAMETERS

| Parameter | Fig. 3 | Fig. 4 | Fig. 5 | Fig. 6 | Fig. 7 | Fig. 8 | Fig. 9 |
|---|---|---|---|---|---|---|---|
| MAXI | 22 | 22 | 22 | 20 | 48 | 48 | 48 |
| MAXJ | 21 | 21 | 21 | 51 | 41 | 41 | 41 |
| ISLT | 11 | 11 | 11 | 10 | 24 | 24 | 24 |
| JLT | 8 | 8 | 8 | 27 | 18 | 18 | 18 |
| JRT | 14 | 14 | 14 | 41 | 24 | 24 | 24 |
| E1 | .01 | .01 | .01 | .01 | .01 | .01 | .01 |
| E2 | .001 | .001 | .001 | .001 | .001 | .001 | .001 |
| R | 1.6 | 1.6 | 1.6 | 1.6 | 1.6 | 1.6 | 1.6 |
| PLT | 0.0 | 20.0 | 20.0 | 25.0 | 0.0 | $2 \times 10^3$ | $2 \times 10^3$ |
| PRT | 0.0 | 20.0 | 30.0 | 25.0 | 0.0 | $2 \times 10^3$ | $2 \times 10^3$ |
| QLT | 0.0 | 20.0 | 20.0 | 25.0 | 0.0 | $2 \times 10^3$ | $2 \times 10^3$ |
| QRT | 0.0 | 20.0 | 30.0 | 25.0 | 0.0 | $2 \times 10^3$ | $2 \times 10^3$ |
| MESH | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| INTL | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| NCRIB | – | – | – | – | 3 | 3 | 3 |
| 6400 CPU TIME | 5 sec. | 4 sec. | 7 sec. | 25 sec. | 54 sec. | 34 sec. | 61 sec. |

## USE OF THE PROGRAM

### INPUT

All input to the program is given in the form of 80-column card images with variables in free format.

The variables MAXI, MAXJ, ISLT, JLT, JRT are specified on the first data card. See Figures 1 and 2.

| Variable | Description |
|---|---|
| MAXI | Number of mesh lines in $\eta$-direction (50 max.). |
| MAXJ | Number of mesh lines in $\xi$-direction (50 max.). |
| ISLT | $\eta$ line to which upper portion of body is mapped. |
| JLT, JRT | Extremities of body on $\xi$ axis. |

The variables E1, E2, R are specified on the next data card.

| Variable | Description |
|---|---|
| E1 | Maximum percent change in values allowed for convergence. $\varepsilon_1$ in Equation (7). |
| E2 | Minimum absolute value of x and included in convergence test. $\varepsilon_2$ in Equation (8). |
| R | Relaxation factor. |

The variables PLT, PRT, QLT, QRT are specified on the succeeding data card.

| Variable | Description |
|---|---|
| PLT | Parameter controlling $\xi$-line attraction to points (JLT,ISLT) and (JLT,ISLT+1). |
| PRT | Parameter controlling $\xi$-line attraction to points (JRT,ISLT) and (JRT,ISLT+1). |
| QLT | Parameter controlling $\eta$-line attraction to points (JLT,ISLT) and (JLT,ISLT+1). |
| QRT | Parameter controlling $\eta$-line attraction to points (JRT,ISLT) and (JRT,ISLT+1). |

The variable MESH, control variable for the type of mesh to be generated, is specified on the next data card. Setting MESH=1 results in a Type 1 mesh; MESH=2 produces a Type 2 mesh. If MESH=2, the variable

NCRIB, the number of grid lines completely encircling the body, is given on the card following. The NCRIB data card is omitted if MESH=1.

The control variable for mesh initialization, INTL, is specified on the succeeding data card. If INTL=0, boundary input data follows immediately. If INTL≠0, the mesh is initialized by reading the previously generated mesh coordinates from input file device TAPE44. The boundary input data cards, if needed, specify the x- and y-coordinates of the boundaries. Each card has an x value followed by a y value; the coordinates are read in the sequence given in the following table. (Refer to Figure 1 for Type 1 mesh and Figure 2 for Type 2 mesh.)

| Type 1 (Figure 1) | Type 2 (Figure 5) |
|---|---|
| 1) A to B | 1) A to B |
| 2) D to C | 2) E to D |
| 3) A to D | 3) A to E |
| 4) B to C | 4) B to D |
| 5) E to F (upper) | 5) K to L |
| 6) E to F (lower) | 6) N to M |

The following table summarizes the user-supplied input needed to generate each type of mesh:

Type 1:

| Card No. | Variables |
|---|---|
| 1 | MAXI, MAXJ, ISLT, JLT, JRT |
| 2 | E1, E2, R |
| 3 | PLT, PRT, QLT, QRT |
| 4 | MESH |
| 5 | INTL |
| 6ff. | boundary input data (if needed) |

Type 2:

| Card No. | Variables |
|---|---|
| 1 | MAXI, MAXJ, ISLT, JLT, JRT |
| 2 | E1, E2, R |
| 3 | PLT, PRT, QLT, QRT |
| 4 | MESH |
| 5 | NCRIB |
| 6 | INTL |
| 7ff | boundary input data (if needed) |

## OUTPUT

The printed output consists of all the input parameters, the number of iterations performed, and ERROR. ERROR is the maximum percent change in the x- and y-values that occurred during the last iteration. This output may be used to check the accuracy of the input data and the convergence of the iteration scheme. The arrays X, Y, SI, TA are written in free format to the output file device TAPE33 in a form suitable for use as initialization input to a subsequent NUMESH run. Initialization in this way usually results in fewer iterations when a mesh is desired which is little changed from the previous one.

| Variables | Description |
|---|---|
| X, Y | Physical coordinates of mesh points. |
| SI, TA | Parameters needed for potential flow solution on generated mesh. |
| I, J | Transformed coordinates of the mesh points. |

## CONTROL CARDS

It is usually desirable to save and catalog the output from NUMESH for future use. The control cards and deck structure needed for execution on the CDC 6700 are as follows:

```
 JØB card
 CHARGE card
 FTN.
 REQUEST,TAPE33,*PF.
+ATTACH,TAPE44,[previous output file name],ID=CXXX.
 LGØ.
 CATALOG,TAPE33,[output file name],ID=CXXX,AC=____.
 7/8/9
  source deck
 7/8/9
  data cards
 6/7/8/9
```

+ This card is omitted if mesh is not to be initialized from a previously generated mesh.

## GRAPHICS

In order to obtain an accurate numerical solution to the partial differential equation under consideration, a suitable mesh must be generated. It is therefore desirable to have a means by which the output from NUMESH may be displayed graphically. Plotting routines with this capability such as IMAGE (documentation will be available in the near future) are available in the Computation, Mathematics, and Logistics Department. For details on the use of these programs, contact Code 1843.

## COMPUTER REQUIREMENTS

NUMESH is designed to run on the CDC 6700 computer system and requires about 75K octal words of storage. It is difficult to estimate the running time because of the large number of factors involved. The computer time required for NUMESH depends on the number of mesh points, the attraction parameters, the convergence criteria, and the geometry of the problem. Execution times for the figures are given in Table 1. NUMESH compiles in under 15 CP seconds on the CDC 6400 processor.

# PROGRAM LISTING

A program listing is given on the following pages.

```
      PROGRAM NUMESH (INPUT,OUTPUT,TAPE33,TAPE44,TAPE5=INPUT)
      COMMON X(22,49),Y(22,49),P(22,49),Q(22,49),MAXI,MAXJ,NIT,AERR,
     1      E1,E2,ISLT,R,SI(22,49),TA(22,49),JLT,JRT,MESH,NCRIB
C
C *** THIS PROGRAM NUMERICALLY GENERATES A BODY-FITTING MESH
C
      CALL INPUT
      IF(MESH.EQ.1) CALL COMPUT1
      IF(MESH.EQ.2) CALL COMPUT2
      CALL OUTPUT
      END
      SUBROUTINE INPUT
      COMMON X(22,49),Y(22,49),P(22,49),Q(22,49),MAXI,MAXJ,NIT,AERR,
     1      E1,E2,ISLT,R,SI(22,49),TA(22,49),JLT,JRT,MESH,NCRIB
C
C *** THIS SUBROUTINE READS INPUT DATA AND COMPUTES THE
C     ATTRACTION FUNCTION VALUES
C
      READ(5,*) MAXI,MAXJ,ISLT,JLT,JRT
      MAXJM1=MAXJ-1
      MAXIM1=MAXI-1
      READ(5,*) E1,E2,R
      PRINT 102,MAXI,MAXJ,ISLT,JLT,JRT,E1,E2,R
  102 FORMAT(/* MAXI=*I5,5X*MAXJ=*I5,5X*ISLT=*I5,5X*JLT=*
     1      I5,5X*JRT=*I5,5X*E1=*F7.3,5X*E2=*F7.3,5X*R=*F5.1/)
      READ(5,*) PLT,PRT,QLT,QRT
      PRINT 100,PLT,PRT,QLT,QRT
  100 FORMAT(/1X*PLT=*F10.2,5X*PRT=*F10.2,5X*QLT=*F10.2,5X*QRT=*F10.2)
      READ(5,*) MESH
      PRINT 105,MESH
  105 FORMAT(/11X*MESH=*I3)
      IF(MESH.EQ.1) GO TO 10
      READ(5,*) NCRIB
      PRINT 200,NCRIB
  200 FORMAT(/11X*NCRIB=*I3)
  10  READ(5,*) INTL
      PRINT 101,INTL
  101 FORMAT(/11X*INTL=*I3)
      ISLTP1=ISLT+1
      IF(INTL.EQ.0) GO TO 20
      DO 9 I=1,MAXI
      DO 9 J=1,MAXJ
  9   READ(44,*) X(I,J),Y(I,J),SI(I,J),TA(I,J),IDUM,JDUM
      PRINT 104
  104 FORMAT(/10X* MESH INITALIZED FROM PREVIOUS RUN*)
      GO TO 30
C
C *** INITIALIZE GRID
C
  20  PRINT 103
  103 FORMAT(/10X* MESH INITIALIZED FROM INPUT DATA*)
      IF(MESH.EQ.2) GO TO 70
      ISLTP1=ISLT+1
      ISLTP2=ISLT+2
      DO 1 J=1,MAXJ
  1   READ (5,*) X(1,J),Y(1,J)
      DO 2 J=1,MAXJ
  2   READ (5,*) X(MAXI,J),Y(MAXI,J)
      DO 3 I=1,ISLT
  3   READ (5,*) X(I,1),Y(I,1)
      X(ISLTP1,1)=X(ISLT,1)
      Y(ISLTP1,1)=Y(ISLT,1)
      DO 4 I=ISLTP2,MAXI
```

```
6     READ (5,*) X(I,1),Y(I,1)
      DO 5 I=1,ISLT
5     READ (5,*) X(I,MAXJ),Y(I,MAXJ)
      X(ISLTP1,MAXJ)=X(ISLT,MAXJ)
      Y(ISLTP1,MAXJ)=Y(ISLT,MAXJ)
      DO 51 I=ISLTP2,MAXI
51    READ (5,*) X(I,MAXJ),Y(I,MAXJ)
      DO 7 J=2,MAXJM1
      DO 7 I=2,MAXIM1
      X(I,J)=X(1,J)
7     Y(I,J)=Y(I,1)
      DO 52 J=JLT,JRT
52    READ (5,*) X(ISLT,J),Y(ISLT,J)
      DO 60 J=JLT,JRT
60    READ (5,*) X(ISLTP1,J),Y(ISLTP1,J)
      GO TO 30
70    ITP=ISLT-NCRIB
      IBM=ISLT+NCRIB+1
      ITPM1=ITP-1
      IBMP1=IBM+1
      DO 71 J=1,MAXJ
71    READ (5,*) X(1,J),Y(1,J)
      DO 72 J=1,MAXJ
72    READ (5,*) X(MAXI,J),Y(MAXI,J)
      DO 73 I=1,ITPM1
73    READ (5,*) X(I,1),Y(I,1)
      DO 731 I=ITP,IBM
731   X(I,1)=Y(I,1)=0.
      DO 732 I=IBMP1,MAXI
732   READ (5,*) X(I,1),Y(I,1)
      DO 74 I=1,ITPM1
74    READ (5,*) X(I,MAXJ),Y(I,MAXJ)
      DO 741 I=ITP,IBM
741   X(I,MAXJ)=Y(I,MAXJ)=0.
      DO 742 I=IBMP1,MAXI
742   READ (5,*) X(I,MAXJ),Y(I,MAXJ)
      DO 8 J=2,MAXJM1
      DO 8 I=2,MAXIM1
      X(I,J)=X(1,J)
8     Y(I,J)=Y(I,1)
      DO 75 J=JLT,JRT
75    READ (5,*) X(ISLT,J),Y(ISLT,J)
      DO 76 J=JLT,JRT
76    READ (5,*) X(ISLTP1,J),Y(ISLTP1,J)
C
C *** INITIALIZE P AND Q ARRAY
C
30    JLT1=JLT
      JRT1=JRT
      IATT1=ISLT
      IATT2=ISLTP1
      ISTART=ISLT+2
      IEND=ISLT-1
      JSTART=2
      JEND=MAXJM1
      IF(MESH.EQ.1) GO TO 18
      JLT1=JLT-2
      JRT1=JRT+2
      IATT1=ISLT-NCRIB+1
      IATT2=ISLT+NCRIB
      ISTART=ISLT+NCRIB+2
      IEND=ISLT-NCRIB-1
      JSTART=JLT
      JEND=JRT
18    DO 6 I=1,MAXI
      DO 6 J=1,MAXJ
```

```
      P(I,J)=Q(I,J)=0.
6     CONTINUE
C
C *** READ ATTRACTION PARAMETERS AND COMPUTE VALUES
C
      JCTR=(JLT+JRT)/2
      ISLTM1=ISLT-1
      ISLTP2=ISLT+2
      DO 11 I=2,IEND
      RI=I
      DO 11 J=JSTART,JCTR
      PLT1=PLT
      IF(J.GT.JLT1) PLT1=-PLT1
      RJ=J
      P(I,J)=P(I,J)+PLT1*EXP(-SQRT((RI-IATT1)**2+(RJ-JLT1)**2))
11    Q(I,J)=Q(I,J)-QLT*EXP(-SQRT((RI-IATT1)**2+(RJ-JLT1)**2))
      DO 12 I=2,IEND
      RI=I
      DO 12 J=JCTR,JEND
      PRT1=PRT
      IF(J.LT.JRT1) PRT1=-PRT1
      RJ=J
      P(I,J)=P(I,J)-PRT1*EXP(-SQRT((RI-IATT1)**2+(RJ-JRT1)**2))
12    Q(I,J)=Q(I,J)-QRT*EXP(-SQRT((RI-IATT1)**2+(RJ-JRT1)**2))
      DO 14 I=ISTART,MAXIM1
      RI=I
      DO 14 J=JSTART,JCTR
      PLT1=PLT
      IF(J.GT.JLT1) PLT1=-PLT1
      RJ=J
      P(I,J)=P(I,J)+PLT1*EXP(-SQRT((RI-IATT2)**2+(RJ-JLT1)**2))
14    Q(I,J)=Q(I,J)+QLT*EXP(-SQRT((RI-IATT2)**2+(RJ-JLT1)**2))
      DO 15 I=ISTART,MAXIM1
      RI=I
      DO 15 J=JCTR,JEND
      PRT1=PRT
      IF(J.LT.JRT1) PRT1=-PRT1
      RJ=J
      P(I,J)=P(I,J)-PRT1*EXP(-SQRT((RI-IATT2)**2+(RJ-JRT1)**2))
15    Q(I,J)=Q(I,J)+QRT*EXP(-SQRT((RI-IATT2)**2+(RJ-JRT1)**2))
      RETURN
      END
      SUBROUTINE COMPUT1
      DIMENSION XO(22,49),YO(22,49)
      COMMON X(22,49),Y(22,49),P(22,49),Q(22,49),MAXI,MAXJ,NIT,AERR,
     1       E1,E2,ISLT,R,SI(22,49),TA(22,49),JLT,JRT,MESH,NCRIB
      MAXIM1=MAXI-1
      MAXJM1=MAXJ-1
      ISLTM1=ISLT-1
      ISLTP1=ISLT+1
      ISLTP2=ISLT+2
      JLTM1=JLT-1
      JRTP1=JRT+1
C
C *** THIS SUBROUTINE COMPUTES THE X AND Y COORINATES OF THE MESH TYPE 1
C
      NIT=0
      DO 10 I=1,MAXI
      DO 10 J=1,MAXJ
      SI(I,J)=TA(I,J)=0.
10    CONTINUE
      DO 1 M=1,50
      NIT=NIT+1
C
C *** SAVE OLD X AND Y VALUES FOR CONVERGENCE CHECK
C
```

```
      DO 2 I=1,MAXI
      DO 2 J=1,MAXJ
      XO(I,J)=X(I,J)
      YO(I,J)=Y(I,J)
2     CONTINUE
C
C *** CALCULATE X AND Y FOR UPPER HALF OF REGION
C
      DO 3 I=2,ISLTM1
      DO 3 J=2,MAXJM1
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+1,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+1,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1       (X(I-1,J+1)-X(I-1,J-1)-X(I+1,J+1)+X(I+1,J-1))/4.
     2       +GA/2./(AL+GA)*(X(I-1,J)+X(I+1,J))
     3       +AJ2*P(I,J)*XX/2./(AL+GA)
     4       +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1       (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+1,J+1)+Y(I+1,J-1))/4.
     2       +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+1,J))
     3       +AJ2*P(I,J)*YX/2./(AL+GA)
     4       +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
3     CONTINUE
C
C *** CALCULATE X AND Y ON SLIT AHEAD OF BODY
C
      I=ISLT
      DO 4 J=2,JLTM1
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+2,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+2,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1       (X(I-1,J+1)-X(I-1,J-1)-X(I+2,J+1)+X(I+2,J-1))/4.
     2       +GA/2./(AL+GA)*(X(I-1,J)+X(I+2,J))
     3       +AJ2*P(I,J)*XX/2./(AL+GA)
     4       +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1       (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+2,J+1)+Y(I+2,J-1))/4.
     2       +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+2,J))
     3       +AJ2*P(I,J)*YX/2./(AL+GA)
     4       +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
      X(ISLTP1,J)=X(I,J)
      Y(ISLTP1,J)=Y(I,J)
      SI(ISLTP1,J)=AJ2*Q(ISLTP1,J)
      TA(ISLTP1,J)=AJ2*P(ISLTP1,J)
4     CONTINUE
C
```

```
C *** CALCULATE X AND Y ON SLIT BEHIND BODY
C
      DO 5 J=JRTP1,MAXJM1
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+2,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+2,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1       (X(I-1,J+1)-X(I-1,J-1)-X(I+2,J+1)+X(I+2,J-1))/4.
     2       +GA/2./(AL+GA)*(X(I-1,J)+X(I+2,J))
     3       +AJ2*P(I,J)*XX/2./(AL+GA)
     4       +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1       (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+2,J+1)+Y(I+2,J-1))/4.
     2       +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+2,J))
     3       +AJ2*P(I,J)*YX/2./(AL+GA)
     4       +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
      X(ISLTP1,J)=X(I,J)
      Y(ISLTP1,J)=Y(I,J)
      SI(ISLTP1,J)=AJ2*Q(ISLTP1,J)
      TA(ISLTP1,J)=AJ2*P(ISLTP1,J)
5     CONTINUE
C
C *** CALCULATE X AND Y BELOW BODY
C
      DO 6 I=ISLTP2,MAXIM1
      DO 6 J=2,MAXJM1
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+1,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+1,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1       (X(I-1,J+1)-X(I-1,J-1)-X(I+1,J+1)+X(I+1,J-1))/4.
     2       +GA/2./(AL+GA)*(X(I-1,J)+X(I+1,J))
     3       +AJ2*P(I,J)*XX/2./(AL+GA)
     4       +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1       (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+1,J+1)+Y(I+1,J-1))/4.
     2       +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+1,J))
     3       +AJ2*P(I,J)*YX/2./(AL+GA)
     4       +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
6     CONTINUE
C
C *** CHECKING CONVERGENCE
C
      AERR=0.
      DO 7 I=1,MAXI
      DO 7 J=JLT,JRT
      IF(ABS(Y(I,J)).LE.E2) GO TO 8
      ERR=ABS((YO(I,J)-Y(I,J))/Y(I,J))
```

```
      IF(ERR.GT.AERR) AERR=ERR
8     IF(ABS(X(I,J)).LE.E2) GO TO 7
      ERR=ABS((XO(I,J)-X(I,J))/X(I,J))
      IF(ERR.GT.AERR) AERR=ERR
7     CONTINUE
      IF(AERR.LE.E1) GO TO 9
1     CONTINUE
9     RETURN
      END
      SUBROUTINE COMPUT2
      DIMENSION XO(22,49),YO(22,49)
      COMMON X(22,49),Y(22,49),P(22,49),Q(22,49),MAXI,MAXJ,NIT,AERR,
     1      E1,E2,ISLT,R,SI(22,49),TA(22,49),JLT,JRT,MESH,NCRIB
      ITP=ISLT-NCRIB
      IBM=ISLT+NCRIB+1
      MAXIM1=MAXI-1
      MAXJM1=MAXJ-1
      ISLTM1=ISLT-1
      ISLTP1=ISLT+1
      ITPM1=ITP-1
      ITPP1=ITP+1
      IBMP1=IBM+1
      IBMM1=IBM-1
      JRTP1=JRT+1
      JLTM1=JLT-1
C
C *** THIS SUBROUTINE COMPUTES THE X AND Y COORINATES OF THE MESH TYPE 2
C
      NIT=0
      DO 200 I=1,MAXI
      DO 200 J=1,MAXJ
      SI(I,J)=TA(I,J)=0.
200   CONTINUE
      DO 1 M=1,50
      NIT=NIT+1
C
C *** SAVE OLD X AND Y VALUES FOR CONVERGENCE CHECK
C
      DO 2 I=1,MAXI
      DO 2 J=1,MAXJ
      XO(I,J)=X(I,J)
      YO(I,J)=Y(I,J)
2     CONTINUE
C
C *** UPDATE DUMMY VALUES
C
      DO 10 J=1,JLTM1
      X(ITP,J)=X(IBMP1,J)
      Y(ITP,J)=Y(IBMP1,J)
10    CONTINUE
      DO 11 J=JRTP1,MAXJ
      X(ITP,J)=X(IBMP1,J)
      Y(ITP,J)=Y(IBMP1,J)
11    CONTINUE
C
C *** CALCULATE X AND Y FOR UPPER HALF OF REGION
C
      DO 3 I=2,ITPM1
      DO 3 J=2,MAXJM1
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+1,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+1,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
```

```
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1      (X(I-1,J+1)-X(I-1,J-1)-X(I+1,J+1)+X(I+1,J-1))/4.
     2      +GA/2./(AL+GA)*(X(I-1,J)+X(I+1,J))
     3      +AJ2*P(I,J)*XX/2./(AL+GA)
     4      +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1      (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+1,J+1)+Y(I+1,J-1))/4.
     2      +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+1,J))
     3      +AJ2*P(I,J)*YX/2./(AL+GA)
     4      +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
3     CONTINUE
C
C *** UPDATE DUMMY VALUES
C
      N=0
      DO 12 I=ITP,IBM
      N=N+1
      X(I,JLTM1)=X(IBMP1-N,JLT)
      X(I,JRTP1)=X(IBMP1-N,JRT)
      Y(I,JLTM1)=Y(IBMP1-N,JLT)
      Y(I,JRTP1)=Y(IBMP1-N,JRT)
12    CONTINUE
C
C *** CALCULATE X AND Y AROUND THE BODY
C
      DO 16 I=ITP,IBM
      IF(I.EQ.ISLT.OR.I.EQ.ISLTP1) GO TO 16
      DO 13 J=JLT,JRT
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+1,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+1,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1      (X(I-1,J+1)-X(I-1,J-1)-X(I+1,J+1)+X(I+1,J-1))/4.
     2      +GA/2./(AL+GA)*(X(I-1,J)+X(I+1,J))
     3      +AJ2*P(I,J)*XX/2./(AL+GA)
     4      +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1      (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+1,J+1)+Y(I+1,J-1))/4.
     2      +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+1,J))
     3      +AJ2*P(I,J)*YX/2./(AL+GA)
     4      +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
13    CONTINUE
16    CONTINUE
C
C *** UPDATE DUMMY VALUES
C
      DO 14 J=1,JLTM1
      X(IBM,J)=X(ITPM1,J)
      Y(IBM,J)=Y(ITPM1,J)
14    CONTINUE
      DO 15 J=JRTP1,MAXJ
      X(IBM,J)=X(ITPM1,J)
```

```
      V(IBM,J)=V(ITPM1,J)
15    CONTINUE
C
C *** CALCULATE X AND Y BELOW BODY
C
      DO 6 I=IBMP1,MAXIM1
      DO 6 J=2,MAXJM1
      XX=(X(I,J+1)-X(I,J-1))/2.
      XE=(X(I-1,J)-X(I+1,J))/2.
      YX=(Y(I,J+1)-Y(I,J-1))/2.
      YE=(Y(I-1,J)-Y(I+1,J))/2.
      AL=XE*XE+YE*YE
      BE=XX*XE+YX*YE
      GA=XX*XX+YX*YX
      AJ2=(XX*YE-XE*YX)**2
      X(I,J)=AL/2./(AL+GA)*(X(I,J+1)+X(I,J-1))-BE/(AL+GA)*
     1       (X(I-1,J+1)-X(I-1,J-1)-X(I+1,J+1)+X(I+1,J-1))/4.
     2       +GA/2./(AL+GA)*(X(I-1,J)+X(I+1,J))
     3       +AJ2*P(I,J)*XX/2./(AL+GA)
     4       +AJ2*Q(I,J)*XE/2./(AL+GA)
      Y(I,J)=AL/2./(AL+GA)*(Y(I,J+1)+Y(I,J-1))-BE/(AL+GA)*
     1       (Y(I-1,J+1)-Y(I-1,J-1)-Y(I+1,J+1)+Y(I+1,J-1))/4.
     2       +GA/2./(AL+GA)*(Y(I-1,J)+Y(I+1,J))
     3       +AJ2*P(I,J)*YX/2./(AL+GA)
     4       +AJ2*Q(I,J)*YE/2./(AL+GA)
      X(I,J)=XO(I,J)+R*(X(I,J)-XO(I,J))
      Y(I,J)=YO(I,J)+R*(Y(I,J)-YO(I,J))
      SI(I,J)=AJ2*Q(I,J)
      TA(I,J)=AJ2*P(I,J)
6     CONTINUE
C
C *** CHECKING CONVERGENCE
C
      AERR=0.
      DO 7 I=2,MAXIM1,2
      DO 7 J=JLT,JRT
      IF(ABS(Y(I,J)).LE.E2) GO TO 8
      ERR=ABS((YO(I,J)-Y(I,J))/Y(I,J))
      IF(ERR.GT.AERR) AERR=ERR
8     IF(ABS(X(I,J)).LE.E2) GO TO 7
      ERR=ABS((XO(I,J)-X(I,J))/X(I,J))
      IF(ERR.GT.AERR) AERR=ERR
7     CONTINUE
      IF(AERR.LE.E1) GO TO 9
1     CONTINUE
C
C *** UPDATE DUMMY VLAUES
C
9     N=0
      DO 17 I=ITP,IBM
      N=N+1
      X(I,JLTM1)=X(IBMP1-N,JLT)
      X(I,JRTP1)=X(IBMP1-N,JRT)
      Y(I,JLTM1)=Y(IBMP1-N,JLT)
      Y(I,JRTP1)=Y(IBMP1-N,JRT)
17    CONTINUE
      JLTM2=JLT-2
      JRTP2=JRT+2
      DO 19 I=ITPP1,IBMM1
      DO 19 J=1,JLTM2
      X(I,J)=0.
 19   Y(I,J)=0.
      DO 18 I=ITPP1,IBMM1
      DO 18 J=JRTP2,MAXJ
      X(I,J)=0.
 18   Y(I,J)=0.
```

```
      RETURN
      END
      SUBROUTINE OUTPUT
      COMMON X(22,49),Y(22,49),P(22,49),Q(22,49),MAXI,MAXJ,NIT,AERR,
     1      E1,E2,ISLT,R,SI(22,49),TA(22,49),JLT,JRT,MESH,NCRIB
C
C *** THIS SUBROUTINE PRINTS OUTPUT DATA AND WRITES X AND Y TO FILE
C
      PRINT 8,NIT,AERR,E1,E2
8     FORMAT(//" ITERATIONS="I5,5X"ERROR="F10.6,5X"E1="F6.3,5X"E2="F6.3)
      DO 30 I=1,MAXI
      DO 30 J=1,MAXJ
      WRITE(33,*) X(I,J),Y(I,J),SI(I,J),TA(I,J),I,J
30    CONTINUE
      RETURN
      END
```

## REFERENCES

1. Thompson, J.F., F.C. Thames and C.W. Mastin, "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Bodies," Journal of Comp. Physics, vol. 15 (1974), pp. 299-319.

2. Thompson, J.F., F.C. Thames, C.W. Mastin and S.P. Shanks, "Use of Numerically Generated Body-Fitted Coordinate Systems for Solution of the Navier-Stokes Equations," Proceedings of AIAA 2nd Comp. Fluid Dynamics Conf., Hartford, Conn. (1975).

3. Thompson, J.F., F.C. Thames, J.K. Hodge, S.P. Shanks, R.N. Reddy and C.W. Mastin, "Solutions of the Navier-Stokes Equations in Various Flow Regimes on Fields Containing Any Number of Arbitrary Bodies Using Boundary-Fitted Coordinate Systems," V. International Conf. on Numerical Methods in Fluid Dynamics, Enschede, The Netherlands (1976).

## ACKNOWLEDGMENT

The author wishes to thank Dr. Henry Haussling for his assistance in the preparation of this report.

## DISTRIBUTION LIST

Copies:

CHONR

| | | |
|---|---|---|
| 1 | 430/R.J. Lundegard | |
| 1 | 430B/M. Cooper | |
| 1 | 438/R.D. Cooper | |

NAVSEA

| | | |
|---|---|---|
| 1 | 03C | |
| 1 | 03F/B. Orleans | |
| 1 | 0351/T. Peirce | |

DTNSRDC

| | | |
|---|---|---|
| 1 | 1500 | W.E. Cummins |
| 1 | 1504 | V. Monacella |
| 1 | 1520 | R. Wermter |
| 1 | 1521 | P. Pien |
| 1 | 1532 | M. Wilson |
| 1 | 1540 | W.B. Morgan |
| 1 | 1542 | B. Yim |
| 4 | 1552 | K.J. Bai |
| | | J. McCarthy |
| | | N. Salvesen |
| | | C.H. Von Kerczek |
| 1 | 1560 | G.R. Hagen |
| 1 | 1572 | M.D. Ochi |
| 1 | 1576 | W. Richardson |
| 1 | 1600 | H.R. Chaplin |
| 1 | 1630 | A.G. Ford |
| 1 | 1800 | G.H. Gleissner |
| 2 | 1802 | F. Frenkiel |
| | | F. Theilheimer |
| 1 | 1805 | E.H. Cuthill |
| 1 | 1809.3 | D. Harris |
| 1 | 1820 | A.W. Camara |
| 1 | 1840 | H.J. Lugt |
| 1 | 1843 | J.W. Schot |
| 25 | 1843 | R.M. Coleman |
| 1 | 1844 | S.K. Dhir |
| 1 | 1850 | T. Corin |
| 1 | 1860 | R.A. Sulit |
| 1 | 1890 | G.R. Gray |

DDC
12

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.